

Commonwealth of Massachusetts
Executive Office of Health and Human Services

EOHHS IT



Service Oriented Architecture
What It Is and
Why It Is the Future of IT

December 5, 2008

Rev. 1.3



Service Oriented Architecture Defined



Service Oriented Architecture (SOA) is a way of creating applications using an Internet approach instead of historically self-contained systems.

This approach emphasizes designing the functionality of systems in smaller, modular ways so that the functionality can be reused by many applications. These functional units, known as 'services', are distributed over a network; a departure from the historical tradition of creating large, monolithic applications. Thus applications that are in different locations can make use of these services.

In addition to the reusable packaging of business and technical functionality, business data that is otherwise duplicated in multiple systems can be positioned once and made available through reusable services, then provided to many 'customer' systems in a standard way.

The success of this approach relies on the use of standards and standard techniques, such as Enterprise Service Bus and Web Service mechanisms, which transcend platform and application-specific architectures.



Service Oriented Architecture

SOA Metaphor



“SOA is technology, of course, and it is architecture, but it isn’t a style of architecture, like mud huts vs. Victorian mansions; it is a scope of architecture, like Building Design vs. Urban Planning. SOA is about turning ad hoc communities of software and process into an integrated economy composed of towns that are part of larger counties that are part of larger states, and so on. SOA is about the design and execution of the master plan, the infrastructure and government and laws that all of an organizations IT entities must follow to enable peaceful, productive commerce all around.”

Joe McKendrick, 2008

**In summary, SOA is part of the transition to the ‘Industrial Age’
of Information Technology and Systems,
and the resolution of fundamental technical choices for the long term.**



Service Oriented Architecture

“The War of Currents”; Direct vs. Alternating



Starting in 1882, Thomas Edison, inventor of the electric light, began the electrification of New York City using Direct Current. In 1886, Nikola Tesla and George Westinghouse proved that Alternating Current and transformers could power Great Barrington, MA.

Despite the ‘first to deploy’ advantage, Alternating Current displaced Edison’s Direct Current and became the standard.

	Direct Current		Alternating Current
Ability to operate lamps and machines	<ul style="list-style-type: none">•Could power lights at a 100 volt standard*•Could power motors at different voltages depending on the size of the motor	=	<ul style="list-style-type: none">•Could power lights at a 100 volt standard*•Could power motors at different voltages depending on the size of the motor
Wiring	<ul style="list-style-type: none">•Used a 3-wire system of varying thicknesses for different voltages. Each voltage required separate wiring. Lights and motors used different voltages	➔	<ul style="list-style-type: none">•Used a 2-wire system to carry all voltages, and used transformers to change the voltage to the required voltage. The thickness of the wires was proportional to the voltage
Distance of uses from sources of generation	<ul style="list-style-type: none">•Required generating stations to be a maximum of 2 miles from the user due to voltage loss during transmission	➔	<ul style="list-style-type: none">•Used transformers to transmit current at very high voltages where loss is minimized, then transform the voltage in steps as it approached the user location. Generating stations could be hundreds of miles from the user
Power Generation	<ul style="list-style-type: none">•Many generation plants were needed, close to users•Different generators were needed for different voltages	➔	<ul style="list-style-type: none">•Fewer, large generators could be deployed at longer distances from users, to service all voltage requirements
Danger to people and structures	<ul style="list-style-type: none">•DC current is relatively safe to people, particularly at the low voltages distributed to users	➔	<ul style="list-style-type: none">•AC current can be fatal at high voltages, and harmful at voltages distributed to users

Displacement of standards takes time. The last vestige of Edison’s DC electrical system in New York City was retired in 2007!

* Today’s standard of 110 volts came from Edison who needed to generate at 110 volts so after transmission loss the voltage to a home would be 100 volts, the standard for light bulbs. Westinghouse matched Edison’s DC generation voltage of 110 volts and so the standard was set.



Service Oriented Architecture

“The War of Currents”; Direct vs. Alternating



IT Services, and Service Oriented Architectures are displacing monolithic and self-contained designs of Information Systems. This trend is driven by the increasing reliability of the network, and the elimination of duplication that isolation promotes.

Despite the ‘first to deploy’ advantage, the Traditional approaches to IT systems are being challenged by SOA.

	Traditional		SOA
IT Infrastructure	<ul style="list-style-type: none">•Self-contained•Locally deployed – local standards•Redundancy with adjacent implementations•Fault Tolerance implemented per-application	➔	<ul style="list-style-type: none">•Distributed architecture•Leverages shared resources•Broad technical and security standards•Shared emergency services
Com-munications	<ul style="list-style-type: none">•Focused on local user community•Periodic and low-speed communications with other applications and organizations	➔	<ul style="list-style-type: none">•Makes no assumptions about the location of users•Uses services and data that are not local
Applications	<ul style="list-style-type: none">•Designed and developed to local or vendor standards•Inconsistent and incomplete designs resulting from time, budget, skills pressures	➔	<ul style="list-style-type: none">•Broad baseline design requirements•Standard patterns and frameworks•Recovery, performance, measurement, auditing designed in by default
Data	<ul style="list-style-type: none">•Local data definitions•Unique transaction sets•Duplicated master data	➔	<ul style="list-style-type: none">•Standardized data definitions•Standardized transaction definitions•Shared reference and master data
Processes Modeled in IT	<ul style="list-style-type: none">•Uniquely designed to local conditions	➔	<ul style="list-style-type: none">•Aligned and reused processes with specialization where necessary
Systems Development	<ul style="list-style-type: none">•Calibrated to available skills or vendor standard	=	<ul style="list-style-type: none">•Limited variation supported by comprehensive methods, tools and techniques
IT Governance	<ul style="list-style-type: none">•Uneven quality processes•Uneven methodology and review	➔	<ul style="list-style-type: none">•Independent quality measurement•Standardized methodology and review

Today's TCP/IP Internet communications standard has spawned a global network of information and business services which is changing the world even more than the productivity gains driven by the 'Computer Revolutions' of the 1950s – 1990s.



Service Oriented Architecture

Seven Principles of SOA



Seven Principles of SOA:

1. SOA Services are open and standards-based.
2. SOA Services are platform-neutral so they behave the same way regardless of how they are written.
3. SOA Services are location-transparent so they appear the same regardless of where they are deployed.
4. SOA Services are peer-to-peer so any service can use any other services to accomplish its tasks.
5. SOA Services are loosely coupled so services can be enhanced without affecting other services.
6. SOA Services are interface-based so the use of a service is solely through the transfer of agreed data.
7. SOA Services are coarsely grained, providing value at a business level, making it more reusable.

The resulting value for customers: Efficiency, reliability and agility.

Adapted from Harriet Fryman, Cognos, 2007



Service Oriented Architecture

SOA Benefits



SOA Benefits to the Business

- Business Process Agility
- IT Cost Reduction
- Reduction in time to new capability
- Scalability
- Business Coordination
- Information Quality

SOA Benefits to Technology

- Standardized System Integration
- Faster Application Development
- Enhanced Functionality Evolution Capabilities
- Systems Deployability
- Process Visibility
- Capability Sharing

Gartner, 2008



Service Oriented Architecture

SOA Methodology



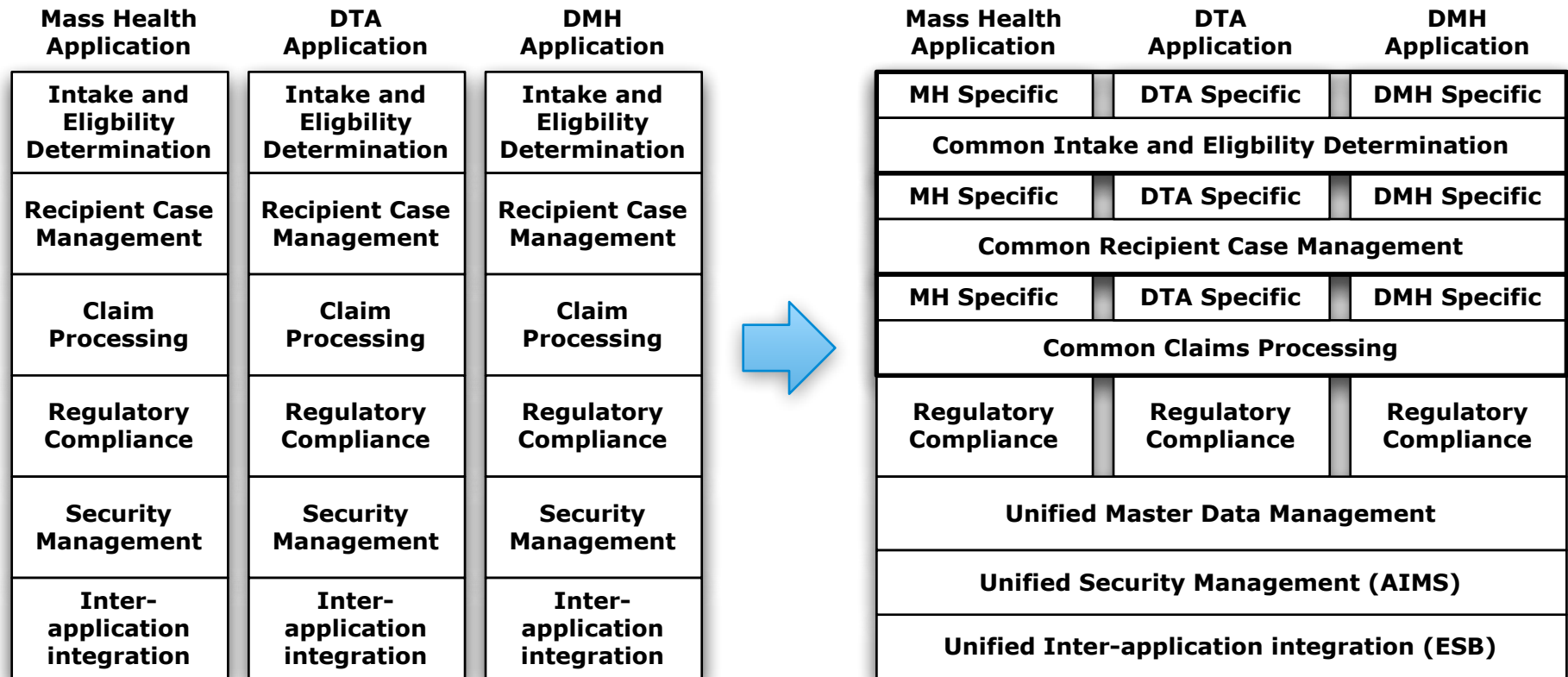
SOA designs are usually combined with methodologies that emphasize many, smaller deliverables. SOA and a methodology such as Unified Process:

- Avoids multi-year projects that deliver only at the end of the project
- Encourages smaller 'services' that deliver complete business or technical functionality in a way that can be reused by more than one project or organization
- Requires disciplined planning at the beginning of a project
- Emphasizes the reduction of risk to the project by delivering and testing portions of a project at a time, including proofs-of-concept of the most difficult parts early in the life of a project
- Independent quality validation throughout the life of a project
- Points of governance for project functional, technical and quality review



Service Oriented Architecture

SOA 'Future State'



Key Attributes of the Services Approach:

- Business processing components can be shared between many applications without precluding some specialization for unique requirements
- Redundant implementations of business processes and Master Data can be rationalized and managed once
- Technical components and services can be mandated across many applications to provide a unified underpinning framework